



# Penetration Test Report (Retest)

Windscribe

Environment: Windscribe  
Dates Assessed: June 24, 2024  
Report Date: June 26, 2024  
Team:

Kyle Burns (kburns@packetlabs.net)  
Richard Rogerson (rogerson@packetlabs.net)  
Denis Kucinic (dkucinic@packetlabs.net)

Table of Contents

1. Risk Level Descriptions..... 3

2. Executive Summary..... 4

3. Approach and Scope ..... 6

4. Technical Findings .....7

4.1 Infrastructure ..... 8

4.1.1 Missing Security Patches .....8

4.1.2 Publicly Accessible Assets: Production Root File System..... 10

4.2 Privacy Review ..... 13

4.2.1 VPN No-Log Policy: End-User Information Logging ..... 13

5. Windscribe Project Conclusion..... 17

6. Methodology ..... 18

6.1 Application Penetration Testing..... 18

6.2 External Infrastructure Penetration Testing ..... 24

# 1. Risk Level Descriptions

## Risk Ratings



Exploitation and discovery of these findings typically require minimal skill and often result in high-privileged access to the affected systems or information. Remediation of critical-risk findings are of high precedence and should not be left unaddressed under any circumstances.



Exploitation of these items can directly lead to the compromise of systems, services or sensitive information. Exploitation is often possible with minimal effort and exploit code is likely to be publicly available or not required. It is recommended that these items be actioned as soon as possible.



Medium-risk findings may lead to a compromise of the environment or disclosure of sensitive information, but may require a significant amount of effort, time and complexity to successfully exploit. Medium risk findings should be actioned in a timely manner.



Low-risk findings have a small impact on the environment and a low likelihood of being exploited. It is generally recommended to address these risks at the lowest priority, occasionally the risk of these findings may be accepted and not actioned due to the limited impact and/or complexity to remediate.



Informational findings are observations made during the assessment which can be addressed with a lower priority. Informational findings typically do not pose a risk to the environment. This may include benign behavior such as bugs and broken functionality.





Remediated findings are findings where the identified vulnerabilities have been determined as fixed with no outstanding risk. Remediated findings do not pose a risk to the environment.

## 2. Executive Summary

### Finding Summary – 2024 Retest

The scope of the retest was focused on all findings from the “2024-06-04 – Windscribe – Penetration Test Report”. The retest was conducted on June 24, 2024, and overall, the environment was found to be at no risk for compromise. The report is based on the current state of the environment as of June 24, 2024. Components of the report have been redacted as to not expose sensitive information about the environment, while still providing an accurate depiction of the results from the Windscribe penetration test and subsequent retest.

Component	Key Findings	Overall Risk Level
	<p>All findings discovered in the “2024-06-04 – Windscribe – Penetration Test Report” were found to have been remediated.</p> <p>Overall, the Windscribe cross-process communication and associated microservices were found to be deployed in a secure manner.</p> <p>As per the no-log policy claims, during the retest, each protocol which was identified to have disclosed client-related information, has since been remediated.</p> <p>Therefore, as of June 24, 2024, in its current state, significant developments in the redaction and prevention of protocol connection logging were observed in the Windscribe node deployment.</p>	


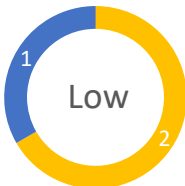
### 2024 Retest Overview

PENETRATION TEST REFERENCE	RISK LEVEL	KEY FINDINGS	RETEST RISK LEVEL
4.1.1 MISSING SECURITY PATCHES		This finding has been remediated.	
4.1.2 PUBLICLY ACCESSIBLE ASSETS: PRODUCTION ROOT FILE SYSTEM		This finding has been remediated.	
4.2.1 VPN NO-LOG POLICY: END-USER INFORMATION LOGGING		This finding has been remediated.	

# Findings Summary – 2024 Penetration Test

Packetlabs was engaged to perform a privacy and infrastructure security assessment of the Windscribe environment. The core objectives of this assessment were to evaluate the security of the Windscribe VPN service and identify potential risk areas. Testing began on May 14, 2024, and was completed on May 28, 2024. During this time, the testing was broken up into logical components based on protocol implementation and Windscribe microservice interactions.

Overall, the environment was found to be at a **low** risk for compromise, with exceptions across **20%** of the OWASP Top 10.

Component	Key Findings	Overall Risk Level
	<p>During testing of the Windscribe cross-communication and associated microservices deployment, no major security risk to the overall environment was identified.</p> <p>Only vulnerabilities relating to the disclosure of sensitive information and the potential exploitation of components using outdated and vulnerable versioning were seen. Compensating controls reduced the likelihood of subsequent exploitation, but ultimately were pointed out as the findings may result in the disclosure of sensitive information.</p> <p>Testing of the Windscribe no-log policy claim was performed which identified several instances of enabled logging across multiple protocol instances. Based on timestamps, most of these were only found to exist for less than a period of forty-eight (48) hours.</p>	

## Strategic Recommendations

- Review the vulnerability management program to ensure patching of software is up to date in a timely manner.
- Conduct annual security assessments to provide continuous monitoring of the Windscribe environment.

# 3. Approach and Scope

## Approach & Methodology

Given the in-scope components outlined in 3.1 below, our approach for this assessment was to segment the testing into logical bundles based on the types of testing. Comprehensive testing was performed on each of the in-scope systems and applications using the methodology outlined in section 6.

### 3.1 Scope

The scope of the assessment was focused the findings disclosed within “**2024-06-04 – Windscribe – Penetration Test Report**”.

This included the ca-023.windscribe.com node along with the security testing and manual source-code review of the Windscribe cross-process communication and microservice stack. The following IPs were considered in scope:

- 181.215.52.172
- 181.215.52.173
- 181.215.52.174
- 181.215.52.175
- 181.215.52.176

Credentials and connection packs for the following protocols were provided and were considered in scope:

- IKEv2 (Strongswan)
- OpenVPN-TCP
- OpenVPN-UDP
- Wireguard
- HTTP Proxy

### 3.2 Constraints & Limitations



Our objective in this Penetration Test was to identify publicly known vulnerabilities residing in systems, applications and infrastructure components. While we have performed extensive testing and analysis, there is no assurance that all vulnerabilities were identified.

Prior to the execution of our testing, we have taken measures to ensure that all of our tools are up to date and are running with the latest feed updates and plugins. This report represents the state of the systems tested at a particular point in time.


# 4. Technical Findings

## Findings Breakdown

Overall, the environment was found to be at no risk for compromise. The overall findings and risk levels have been outlined in the table below.

Component	Key Findings	Overall Risk Level
 External Infrastructure	<b>Closed Findings</b> <ul style="list-style-type: none"><li>✓ 4.1.1 Missing Security Patches</li><li>✓ 4.1.2 Publicly Accessible Assets: Production Root File System</li></ul>	REMIEDIATED
 Privacy Review	<b>Closed Findings</b> <ul style="list-style-type: none"><li>✓ 4.2.1 VPN No-Log Policy: End-User Information Logging</li></ul>	REMIEDIATED

## OWASP Overview

	OWASP Top 10 (2021)	Findings
 Web Application	A1 – Broken Access Control	No
	A2 – Cryptographic Failures	No
	A3 – Injection	No
	A4 – Insecure Design	No
	A5 – Security Misconfiguration	No
	A6 – Vulnerable and Outdated Components	No
	A7 – Identification and Authentication Failures	No
	A8 – Software and Data Integrity Failures	No
	A9 – Security Logging and Monitoring Failures	No
	A10 – Server-Side Request Forgery	No

## 4.1 Infrastructure

### 4.1.1 Missing Security Patches



**INDUSTRY REFERENCE**  
CSC7: Continuous  
Vulnerability Management

**IMPACT**  
Unauthorized access

**ROOT CAUSE**  
Missing security  
patches

Two packages were found missing security patches on infrastructure that is part of the Windscribe environment. As a result, both the internal and external infrastructure are open to multiple security vulnerabilities. Overall, the vulnerable unpatched software discovered was related to:

- Sonatype Nexus Repository
- Redis Metrics Exporter

## Supporting Evidence

### Redis Metrics Exporter

- ca-023.windscribe.com (Redis Metrics Exporter) was found to be using google.golang.org/protobuf (< 1.33.0), which has the associated CVE-2024-24786 assigned.

### Sonatype Nexus Repository

- The repository hosted on \*.windscribe.net was found to be running an outdated version making it vulnerable to path traversal (CVE-2024-4956). Due to the use of Cloudflare, the vulnerability could not be exploited.

## Recommendation

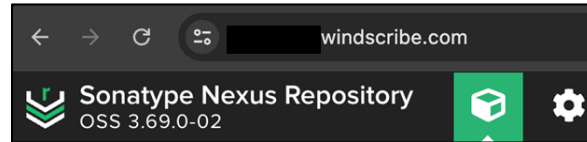
- Apply the latest security patches and validate that missing patches are part of a recurring patch schedule.



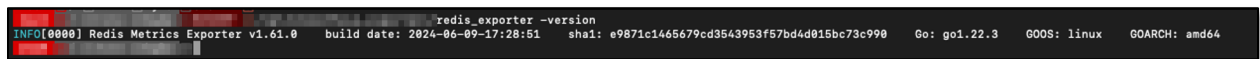
## Supporting Evidence – Retest

Both assets of the Windscribe infrastructure were found to have been updated to the latest version.

Sonatype Nexus Repository OSS 3.69.0-02



Redis Metrics Exporter v1.61.0



## Affected Assets

- ✓ ca-023.windscribe.com (Redis Metrics Exporter)
- ✓ \*.windscribe.net (Sonatype Nexus Repository)

# 4.1.2 Publicly Accessible Assets: Production Root File System



**INDUSTRY REFERENCE**  
OWASP Top 10: A2  
Cryptographic Failures

**IMPACT**  
Disclosure of sensitive  
information

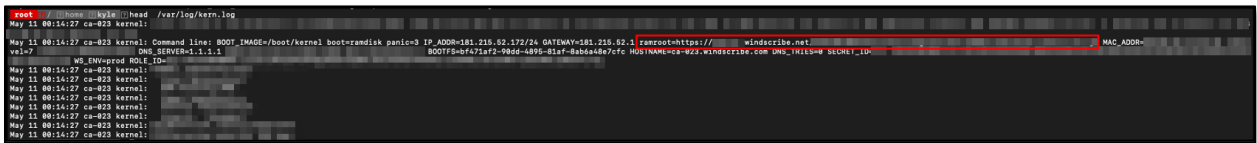
**ROOT CAUSE**  
Insecure configuration

Based on logs from /var/log/kern.log on the May 11, 2024 deployment. The Kernel / Initramfs build process uses the root file system hosted on a private Sonatype Nexus Repository. Normally, accessing the endpoint \*.windscribe.net requires authorization, however, by directly requesting the `redacted.tar.gz` file, the latest deployment can be downloaded from an unauthenticated perspective.

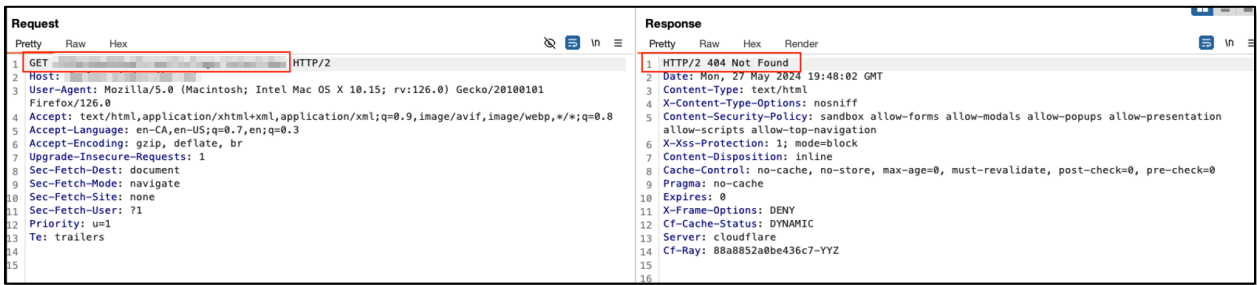
Given the directory, a part of the identified URI contains a 13-character long alpha-numerical string, it is unlikely that the endpoint would be discovered. However, the use of 404 in response to a non-existent location instead of the generic 401 could potentially be used to derive the directory structure and subsequently download the `rootfs` file.

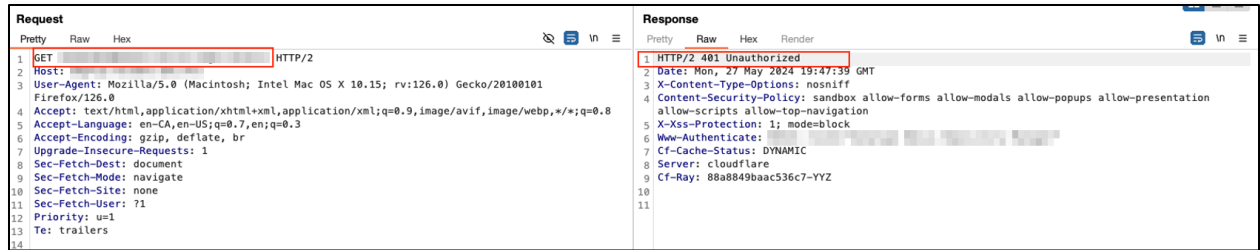
## Supporting Evidence

The Initramfs build is pulled from the staging URL ([https://\\*.windscribe.net/redacted](https://*.windscribe.net/redacted)).

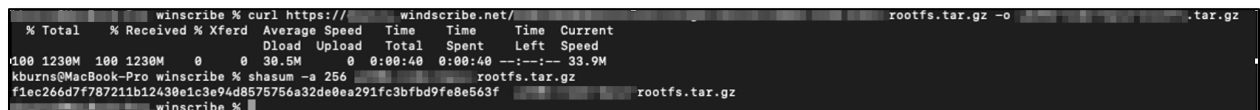


Externally, the endpoint requires no authentication to download the rootfs file, however the likelihood of finding the directory is not trivial. One caveat to this exists, due to the HTTP server's response code, containing 404 for non-existent endpoints, and 401 for valid ones, the endpoint could be slowly brute-forced.





Given the increased likelihood, the `redacted.tar.gz` file can be downloaded without authentication, disclosing sensitive information present on the file system.



Sensitive information such as the salted SHA-512 hash for the redacted user, along with standard users (redacted, redacted) are exposed. Lastly, binaries, including the one found in `/opt/redacted/bin/redacted` can be reverse-engineered to disclose sensitive information or provide further microservice context.

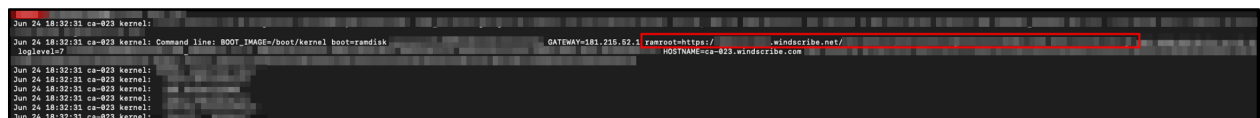


## Recommendation

Restrict access to all endpoints present on the deployment server and return consistent response codes.

## Supporting Evidence – Retest

Deployment now occurs on the \*.windscribe.net asset which utilizes mutual TLS (mTLS) for client-side verification.



Therefore, any attempt to request the rootfs file without the provisioned certificate will fail.

```
* curl: (6) OpenSSL SSL read: OpenSSL/3.1.5: error:0A00045C:SSL routines::tlsv13 alert certificate required, errno 0
```

## Affected Assets

- ✓ \*.windscribe.net/redacted.tar.gz

## 4.2 Privacy Review

### 4.2.1 VPN No-Log Policy: End-User Information Logging



**INDUSTRY REFERENCE**  
OWASP Top 10: A5  
Security Misconfiguration

**IMPACT**  
Sensitive information  
disclosure

**ROOT CAUSE**  
Insecure configuration

The Windscribe product operates on a no-log policy, which allows users to effectively use the service without the potential for client-origin data to be logged or leaked to unwanted third-parties. Windscribe requested a privacy review to validate the effectiveness of the claims. As a result of the review, it was identified that certain configurations applied to the environment logged sensitive information of the end user.

## Supporting Evidence

### IKEv2 (StrongSwan)

The IKEv2 service was found to store logs within volatile memory for at least 18 hours. The associated logs contain the origin IP of the connecting users and are generated on the initiation, establishment, and disconnection of the VPN session.

```
May 23 20:28:21 charon[17698]: 217[ENC] generating INFORMATIONAL response 7 [ ]
May 23 20:28:21 charon[17698]: 217[IKE] IKE_SA deleted
May 23 20:28:21 charon[17698]: 217[ENC] deleting IKE_SA ikev2-mschapv2[36696] between 181.215.52.172[Chica-023.windscribe.com]...
May 23 20:28:21 charon[17698]: 217[ENC] parsed INFORMATIONAL request 7 [ D ]
May 23 20:28:21 charon[17698]: 168[ENC] generating INFORMATIONAL response 6 [ D ]
May 23 20:28:21 sudo[1486925]:
May 23 20:28:21 sudo[1486925]:
May 23 20:28:21 charon[17698]: 168[ENC] closing CHILD_SA ikev2-mschapv2[4] with SPIs c13fcc16_i (15963 bytes) e766c898_o (14369 bytes) and
May 23 20:28:21 charon[17698]: 168[ENC] parsed INFORMATIONAL request 6 [ D ]
May 23 20:27:19 charon[17698]: 57[ENC] generating IKE_AUTH response 5 [ AUTH CPRP(ADDR DNS) SA TSI Tsr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) ]
May 23 20:27:19 sudo[1486821]:
May 23 20:27:19 sudo[1486821]:
May 23 20:27:19 charon[17698]: 57[IKE] CHILD_SA ikev2-mschapv2[4] established with SPIs c13fcc16_i e766c898_o and TS 0.0.0.0/0...
May 23 20:27:19 charon[17698]: 57[ENC] generating INFORMATIONAL response 5 [ AUTH CPRP(ADDR DNS) SA TSI Tsr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) ]
May 23 20:27:19 charon[17698]: 57[ENC] parsed IKE_AUTH request 5 [ AUTH CPRP(ADDR DNS) SA TSI Tsr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) N(ADD_4_ADDR) ]
May 23 20:27:19 charon[17698]: 52[ENC] generating IKE_AUTH response 4 [ EAP/SUCC ]
May 23 20:27:19 charon[17698]: 52[ENC] parsed IKE_AUTH request 4 [ EAP/RES/MSCHAPV2 ]
May 23 20:27:19 charon[17698]: 34[ENC] generating IKE_AUTH response 3 [ EAP/REQ/MSCHAPV2 ]
May 23 20:27:19 charon[17698]: 34[ENC] parsed IKE_AUTH request 3 [ EAP/RES/MSCHAPV2 ]
May 23 20:27:19 charon[17698]: 15[ENC] generating IKE_AUTH response 2 [ EAP/REQ/MSCHAPV2 ]
May 23 20:27:19 charon[17698]: 15[ENC] parsed IKE_AUTH request 2 [ EAP/RES/ID ]
May 23 20:27:19 charon[17698]: 29[ENC] generating IKE_AUTH response 1 [ EF(1/3) ]
May 23 20:27:19 charon[17698]: 29[ENC] generating IKE_AUTH response 1 [ EF(2/3) ]
May 23 20:27:19 charon[17698]: 29[ENC] generating IKE_AUTH response 1 [ EF(3/3) ]
May 23 20:27:19 charon[17698]: 29[ENC] splitting IKE message (3856 bytes) into 3 fragments
May 23 20:27:19 charon[17698]: 29[ENC] generating IKE_AUTH response 1 [ IDr CERT CERT AUTH EAP/REQ/ID ]
May 23 20:27:19 charon[17698]: 29[ENC] parsed IKE_AUTH request 1 [ IDr CERTREQ N(MOBIKE_SUP) CPRQ(ADDR DNS NINS SRV ADDR6 DNS6 SRV6) SA TSI Tsr ]
May 23 20:27:19 charon[17698]: 29[ENC] received fragment #2 of 2, reassembled fragmented IKE message (912 bytes)
May 23 20:27:19 charon[17698]: 29[ENC] parsed IKE_AUTH request 1 [ EF(2/2) ]
May 23 20:27:19 charon[17698]: 238[ENC] generating IKE_SA_INIT response 1 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP) N(CHOLESS_SUP) N(MULT_AUTH) ]
May 23 20:27:19 charon[17698]: 238[ENC] received unknown vendor ID:
May 23 20:27:19 charon[17698]: 238[ENC] received IKE_SA_INIT request
```

In addition to the volatile memory, two files `/var/log/auth.log` and `/var/log/auth.log.1` were found which contained the same logs from the `strongswan` service, but with the exception of `/var/log/auth.log.1` existing for a longer period of time, with its creation date (May 20, 2024).

```

May 28 17:21:20 ca-023 86[IKE] is initiating an IKE_SA
May 28 17:21:20 ca-023 172[IKE] IKE_SA ikev2-mschapv2[63153] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 28 17:23:54 ca-023 85[IKE] deleting IKE_SA ikev2-mschapv2[63153] between 181.215.52.172[CN=ca-023.windscribe.com]...
May 28 17:23:56 ca-023 217[IKE] is initiating an IKE_SA
May 28 17:23:56 ca-023 203[IKE] IKE_SA ikev2-mschapv2[63154] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 28 17:24:11 ca-023 132[IKE] deleting IKE_SA ikev2-mschapv2[63154] between 181.215.52.172[CN=ca-023.windscribe.com]...
May 28 17:24:29 ca-023 143[IKE] is initiating an IKE_SA
May 28 17:24:30 ca-023 234[IKE] IKE_SA ikev2-mschapv2[63155] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 28 17:24:33 ca-023 17[IKE] deleting IKE_SA ikev2-mschapv2[63155] between 181.215.52.172[CN=ca-023.windscribe.com]...
May 23 16:05:45 ca-023 201[IKE] is initiating an IKE_SA
May 23 16:05:46 ca-023 119[IKE] IKE_SA ikev2-mschapv2[35758] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 23 16:08:06 ca-023 230[IKE] deleting IKE_SA ikev2-mschapv2[35758] between 181.215.52.172[CN=ca-023.windscribe.com]...
May 23 16:08:11 ca-023 221[IKE] is initiating an IKE_SA
May 23 16:08:12 ca-023 30[IKE] IKE_SA ikev2-mschapv2[35759] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 23 16:08:55 ca-023 251[IKE] deleting IKE_SA ikev2-mschapv2[35759] between 181.215.52.172[CN=ca-023.windscribe.com]...
May 23 16:08:59 ca-023 237[IKE] is initiating an IKE_SA
May 23 16:09:00 ca-023 231[IKE] IKE_SA ikev2-mschapv2[35760] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 23 16:13:47 ca-023 209[IKE] deleting IKE_SA ikev2-mschapv2[35760] between 181.215.52.172[CN=ca-023.windscribe.com]...
May 23 20:27:19 ca-023 238[IKE] is initiating an IKE_SA
May 23 20:27:19 ca-023 57[IKE] IKE_SA ikev2-mschapv2[36696] established between 181.215.52.172[CN=ca-023.windscribe.com]..
May 23 20:28:21 ca-023 217[IKE] deleting IKE_SA ikev2-mschapv2[36696] between 181.215.52.172[CN=ca-023.windscribe.com]...

```

```

File: auth.log.1
Size: 5259418      Blocks: 10288      IO Block: 4096      regular file
Device: fc00h/64512d Inode: 393927      Links: 1
Access: (0640/-rw-r-----)  Uid: (  0/      root)   Gid: (  4/      adm)
Access: 2024-05-28 17:25:10.619921396 +0000
Modify: 2024-05-27 15:59:02.047588751 +0000
Change: 2024-05-27 16:00:01.083871592 +0000
Birth: 2024-05-20 18:00:05.831138668 +0000

```

Based on the above, customers of Windscribe that use the IKEv2 (strongswan) protocol may be at risk for leaking their IP, session start, and session stop times.

## OpenVPN

Other services such as OpenVPN (TCP) and OpenVPN (UDP) were seen to implement significant work into limiting the logging of any sensitive information relating to the connecting client.

Only the username of the associated account was found to be temporarily stored while an active session is established. This was found due to the openvpn `status` configuration directive:

```

(`/opt/openvpn/etc/server_udp.conf`). (`status
/var/log/windscribe/openvpn/openvpn_udp.status 1`) and
(`/opt/openvpn/etc/server_tcp.conf`). (`status
/var/log/windscribe/openvpn/openvpn_tcp.status 1`).

```

Additionally, directives within `/opt/openvpn/etc/server\_udp.conf` and `/opt/openvpn/etc/server\_tcp.conf` show an active session for the specific user account (username). However, from a privacy perspective, given it is deleted on termination of the session, this configuration is not deemed a concern.

A potential area of concern relates to the user temporary configuration location for both TCP (`/opt/openvpn/tmp/tcp/`) and UDP (`/opt/openvpn/tmp/udp/`). Both server\_udp.conf and server\_tcp.conf use the directive `client-config-dir` to store OpenVPN client configurations and is done by generating a file associated with the username.

Based on the file creation date (2024-05-11 00:50:11.212348319 +0000), this could be used to leak two sets of information:

1. The first time the user used OpenVPN on the node.
2. The last time the user established a connection using OpenVPN on the node.

```
stat /opt/openvpn/tmp/tcp/
File: /opt/openvpn/tmp/tcp/
Size: 35      Blocks: 8      IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 265360    Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2024-05-28 20:04:47.246370246 +0000
Modify: 2024-05-27 18:55:22.398928150 +0000
Change: 2024-05-27 18:55:22.398928150 +0000
Birth: 2024-05-11 00:50:11.212348319 +0000
stat /opt/openvpn/tmp/tcp/
File: /opt/openvpn/tmp/tcp/
Size: 35      Blocks: 8      IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 265360    Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2024-05-28 20:31:48.686162401 +0000
Modify: 2024-05-28 20:31:48.686162401 +0000
Change: 2024-05-28 20:31:48.686162401 +0000
Birth: 2024-05-11 00:50:11.212348319 +0000
```

For instance, the Windscribe users `redacted` and `redacted` (Packetlabs provisioned account) are accounts that have used the OpenVPN service.

Windscribe Stack

In most instances, service logs only described communication between inner microservices. However, service logs associated with the `redacted` service were found to leak multiple sensitive pieces of information that could be used to derive user identity or parts of it:

- 1. The OpenVPN username and password credentials, session start, and stop are disclosed within the `redacted` scoped requests.
- 2. Wireguard Pre-shared Key (PSK) and user\_id integer during initial authentication.
- 3. IKEv2 (StrongSwan), the username, challenge response keys, client initial connection, and disconnect time.

This information is subsequently stored on disk in `/var/log/debug` and is kept on disk for a minimum of a 10-hour period. Other logs were discovered to contain the same Windscribe stack debugging logs, but for a longer period (23-hours). The file was located on disk at `/var/log/debug.1`. The cause for this file's existence was not determined.

```
journalctl --no-hostname --boot --unit --no-pager
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.648207","event":"relaying message","destination":"","msg_id":"","scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.648207","event":"unrelayed relay message payload","msg_id":"","scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.648207","event":"parsed message from","over_websocket":{"message":{"Version":0,"rad":"","Payload":"XXXXXXXX","msg_id":"","response.171538788866","scope":""}}},"scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.638817","event":"waiting for next","send_queue_message":{"scope":"","1716454589378","scope":""}}},"scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.638817","event":"dropping sent","message_from_queue":{"msg_id":"","1716454589378","scope":""}}},"scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.637979","event":"relaying message to","over_websocket":{"message":{"Version":0,"Payload":"XXXXXXXX","msg_id":"","marvin_connect_request.1716454589378","scope":""}}},"scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.637979","event":"relaying message","destination":"","msg_id":"","1716454589378","queue":"","scope":""}}},"scope":""}}
May 26 16:41:47 {"data":{"2024-05-24 16:41:47.637979","event":"parsed message","message":{"Version":0,"rad":"","Payload":"XXXXXXXX","msg_id":"","marvin_connect_request.1716454589378","queue":"","scope":""}}},"scope":""}}
May 26 16:41:46 {"data":{"2024-05-24 16:41:46.651459","event":"relaying message","destination":"","msg_id":"","171538788866","scope":""}}},"scope":""}}
```

Wireguard

Besides the information discovered as a part of the Windscribe microservices socket communication, no instances of long-term storage of customer information were disclosed when using this protocol.

HTTP Proxy (nghttpx)

No instances of long-term storage of customer information were discovered when using this protocol.

## Recommendation

Reduce the verbosity of the Windscribe service logs by removing the JSON `redacted` key contents. For other services, depending on the use case, redirect client-related data to `/dev/null`.

## Supporting Evidence – Retest

The IKEv2 (StrongSwan) (strongswan.service) was found to no longer log any sensitive information used to identify the connecting user.

```
cat /opt/strongswan/etc/strongswan.d/charon-logging.conf
charon {
  syslog {
    daemon {
      default = -1
    }
  }
}
charon-systemd : charon {
  journal {
    default = -1
  }
}
charon {
  filelog {
    charon {
      default = -1
    }
  }
}
```

Both OpenVPN-UDP and OpenVPN-TCP no longer can be used to derive relevant user information that had previously been found to disclose when and who used the service. This was found to be achieved by the use of dynamically generated file names with no association to the end user, and the eventual deletion of the associated file.

```
ls -la
total 8
drwx----- 2 openvpn openvpn 4096 Jun 24 17:04 .
drwx----- 4 openvpn openvpn 4096 Jun 21 06:43 ..
-rw----- 1 root root 0 Jun 24 17:04 openvpn_acf_1ce6a4ee67f0378057ea942db6bf622a.tmp
stat openvpn_acf_1ce6a4ee67f0378057ea942db6bf622a.tmp
File: openvpn_acf_1ce6a4ee67f0378057ea942db6bf622a.tmp
Size: 0 Blocks: 0 IO Block: 4096 regular empty file
Device: fb00h/64256d Inode: 29763 Links: 1
Access: (0600/-rw-----) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2024-06-24 17:04:07.911310168 +0000
Modify: 2024-06-24 17:04:07.827309821 +0000
Change: 2024-06-24 17:04:07.827309821 +0000
Birth: 2024-06-24 17:04:07.827309821 +0000
```

Lastly, microservice communication facilitated through the `redacted` service no longer logs any sensitive information that could be used to disclose information relating to the end user.

## Affected Assets

- ✓ ca-023.windscribe.com
  - Wireguard
  - OpenVPN (TCP)
- OpenVPN (UDP)
- IKEv2 (StrongSwan)



# 5. Windscribe Project Conclusion

The Windscribe penetration test was a multi-objective penetration test with focuses on two areas:

- The overall security of the Windscribe decentralized node, consisting of source-code security review against the microservice, their inner-process communication, and the other components that enabled Windscribe to operate.
- Validation of the no-log policy to ensure end user anonymity.

Overall, the penetration test against the Windscribe stack yielded minimal security concerns. During the retest, all findings that posed any potential risk were promptly remediated.

With regards to the discoveries related to the no-log policy audit, after a discussion with the Windscribe engineering team, it was identified that some components of the finding were due-in-part to configurational changes being improperly deployed on the designed testing node.

Although this does not cover every discovery noted within the finding, subsequent configurational, and service changes were seen to have been deployed during the retest to resolve the issues identified within the original penetration testing report. Based on the changes deployed during the retest and the source-code reviewed during the original penetration test, it's evident that notable development work has been implemented as a part of the Windscribe microservice stack to reduce, and/or prevent the disclosure of end user information.

# 6. Methodology

## 6.1 Application Penetration Testing

Our security testing methodology is derived from the OWASP Top 10:2021 and has been enhanced with current threats and our overall experience in the industry. Our methodology is comprehensive and has been broken up based on which areas can be tested with automation and those which require extensive manual testing.

Phase	Tasks Completed	Manual	Automated
Information Gathering	Conduct search engine discovery and reconnaissance for information leakage	✓	✓
	Fingerprint web server	✓	✓
	Review web server metafiles for information leakage	✓	✓
	Enumerate applications on web servers	✓	✓
	Review webpage comments and metadata for information leakage	✓	✓
	Identify application entry points	✓	✓
	Identify technologies (e.g., web applications, frameworks, or CMS platforms) used	✓	✓
	Map visible content and perform automated spidering of referenced content	✓	✓
	Test for debug parameters	✓	✓
	Discover hidden & default content	✓	✓
Discovery	Configuration and Deploy Management Testing		
	Test network/infrastructure configuration	✓	✓
	Test application platform configuration	✓	✓
	Test file extensions handling for sensitive information	✓	✓
	Analyze backup and unreferenced files for sensitive information	✓	✓
	Enumerate Infrastructure and application admin interfaces	✓	✓
	Test HTTP methods	✓	✓
	Test HTTP strict transport security	✓	✓
	Test RIA cross-domain policy	✓	✓
	Test for web server vulnerabilities	✓	✓
	Testing for vulnerabilities in third-party applications (e.g. WordPress, Joomla, Drupal, SharePoint)	✓	✓
	Test File Permission	✓	-
	Test for Subdomain Takeover	✓	-
	Test Cloud Storage	✓	-
	Identity Management Testing		
	Test role definitions	✓	-

Phase	Tasks Completed	Manual	Automated
	Test user registration process	✓	-
	Test account provisioning process	✓	-
	Testing for account enumeration and guessable user account	✓	-
	Testing for weak or unenforced username policy	✓	-
	Test permissions of guest/training accounts	✓	-
	Test account suspension/resumption Process	✓	-
	Authentication Testing		
	Testing for credentials transported over an encrypted channel	✓	✓
	Testing for default credentials	✓	✓
	Testing for a weak lockout mechanism	✓	-
	Testing for bypassing authentication schema	✓	-
	Test remember password functionality	✓	-
	Testing for browser cache weakness	✓	✓
	Testing for weak password policy	✓	-
	Testing for weak security question/answer	✓	-
	Testing for weak password change or reset functionalities	✓	-
	Testing for weaker authentication in alternative channel	✓	✓
	Authorization Testing		
	Testing directory traversal/file include	✓	-
	Testing for bypassing authorization schema	✓	-
	Testing for privilege escalation	✓	-
	Testing for insecure direct object references	✓	-
	Session Management Testing		
	Testing for bypassing session management schema	✓	-
	Testing Session Management Schema	✓	-
	Analyze cookies attributes (e.g., HttpOnly, Secure flags and scope)	✓	-
	Testing for session fixation	✓	-
	Testing for cross-site request forgery	✓	-
	Testing for logout functionality	✓	-
	Test session timeout	✓	-
	Testing for session puzzling	✓	-
	Persistent cookies	✓	-
	Test tokens for predictability	✓	-
	Check for insecure transmission of session tokens	✓	-
	Input Validation Testing		
	Fuzz all input parameters	✓	✓
	Testing for Format String Injection	✓	✓

Phase	Tasks Completed	Manual	Automated
	Testing for HTTP Incoming Requests	✓	✓
	Testing for Server-Side Request Forgery	✓	✓
	Testing for reflected cross-site scripting	✓	✓
	Testing for stored cross-site scripting	✓	✓
	Testing for HTTP verb tampering	✓	✓
	Testing for HTTP parameter pollution	✓	✓
	Testing for HTTP splitting/smuggling	✓	✓
	Testing for SQL injection (Oracle, MySQL, MsSQL, PostgreSQL, Microsoft Access, NoSQL)	✓	✓
	Testing for LDAP injection	✓	✓
	Testing for ORM injection	✓	✓
	Testing for XML injection	✓	✓
	Testing for SSI injection	✓	✓
	Testing for XPath injection	✓	✓
	Testing for IMAP/SMTP injection	✓	✓
	Testing for code injection	✓	✓
	Testing for local file inclusion	✓	✓
	Testing for remote file inclusion	✓	✓
	Testing for command injection	✓	✓
	Testing for native software flaws (buffer overflow, integer bugs, format strings)	✓	✓
	Testing for incubated vulnerabilities	✓	✓
	Testing for open redirection	✓	✓
	Testing for SOAP injection	✓	✓
	Error Handling		
	Analysis of error codes	✓	✓
	Analysis of stack traces	✓	✓
	Cryptography		
	Testing for weak SSL/TLS ciphers, insufficient transport layer protection	✓	✓
	Testing for padding oracle	✓	✓
	Testing for sensitive information sent via unencrypted channels	✓	✓
	Testing for CBC bit flipping	✓	✓
	Testing for hash length extension	✓	✓
	Business Logic Testing		
	Identify the logic attack surface	✓	-
	Test business logic data validation	✓	-
	Test the ability to forge requests	✓	-
	Test integrity checks	✓	-

Phase	Tasks Completed	Manual	Automated
	Test for process timing (race conditions, TOCTOU)	✓	-
	Testing for the circumvention of workflows (e.g., payments)	✓	-
	Test defenses against application misuse	✓	-
	Test upload of unexpected file types	✓	-
	Test upload of malicious files	✓	-
	Analyze SSL responses for caching of sensitive content	✓	-
	Analyze content for sensitive data in URL parameters	✓	-
	Testing for reliance on client-side input validation	✓	-
	Testing of trust boundaries	✓	-
	Client-Side Testing		
	Testing for DOM-based cross-site scripting	✓	✓
	Testing for JavaScript execution	✓	✓
	Testing for HTML injection	✓	✓
	Testing for client-side open redirection	✓	✓
	Testing for CSS injection	✓	✓
	Testing for client-side resource manipulation	✓	✓
	Test cross-origin resource sharing	✓	✓
	Testing for cross-site flashing	✓	✓
	Testing for clickjacking	✓	✓
	Testing WebSockets	✓	-
	Test web messaging	✓	-
	Test local storage	✓	-
	Testing of thick-client components (Java, ActiveX, Flash)	✓	-
	Miscellaneous: WordPress		
	Test for outdated plugins	✓	-
	Test for XMLRPC exposure	✓	-
	Test for exposed admin portal	✓	-
	Miscellaneous: JavaScript		
	Test for overly permissive Content Security Policy (CSP)	✓	-
	Test for subresource integrity checks	✓	-
	Testing for linking to third-party code	✓	-
	Testing for advertisement and analytics on critical flows	✓	-
	Testing for critical flows isolation	✓	-
	Leverage findings from previous phases in order to expand foothold in the environment	✓	-
	Miscellaneous: JWT		
	Testing for insufficient expiration on logout	✓	-
	Testing for token information disclosure	✓	-
	Testing for token storage on client Side	✓	-
	Testing for weak token security	✓	-

Phase	Tasks Completed	Manual	Automated
	Testing for insufficient signature validation	✓	-
	Testing for substitution attacks	✓	-
	Miscellaneous: OAuth		
	Missing CSRF protection	✓	-
	Testing for improper usage of implicit grant type	✓	-
	Testing for flawed redirect_uri validation	✓	-
	Miscellaneous: WebRTC		
	Test for SIP Vulnerabilities	✓	-
	Test for Registration & Session Hijacking	✓	-
	Test for replay attacks	✓	-
	Miscellaneous: GraphQL: Reconnaissance		
	Port Scan to Identity Open Web Application Ports	✓	✓
	Analyze Invalid/Valid Server Responses	✓	✓
	Inspect __typename Field for GraphQL Implementations	✓	✓
	Identify Development Endpoints	✓	✓
	Search for server-level vulnerabilities on MITRE's CVE database	✓	✓
	Search for server-level security features on GraphQL Threat Matrix	✓	✓
	Send an introspection query and document all available queries, mutations, and subscriptions	✓	✓
	Visualize introspection query response with GraphQL Voyager	✓	✓
	Review the API's SDL file for Bi-Directional Relationships	✓	✓
	GraphQL: Denial of Service		
	Testing for circular queries or mutations	✓	-
	Testing for circular fragments	✓	-
	Testing for field duplication	✓	-
	Testing for alias overloading	✓	-
	Testing for directive overloading	✓	-
	Testing for array-based or alias-based query batching	✓	-
	Testing for object-limit overriding in API pagination arguments (e.g. filter, max, limit, total)	✓	-
	GraphQL: Information Disclosure		
	Extract GraphQL Schema via Field Stuffing/Enumeration	✓	✓
	Fuzz Queries to Inspect Verbose Errors	✓	✓
	Identify Query Tracing in GraphQL Responses	✓	-
	Examine Query Response for PII Disclosure	✓	-
	Test for GraphQL Debug Mode	✓	✓
	GraphQL: Authentication and Authorization		

Phase	Tasks Completed	Manual	Automated
	Attempt to Access API without Authentication Headers	✓	-
	Attempt to Access Restricted Fields by Using Alternate Paths	✓	-
	Attempt to Access API by transforming GET and POST Methods	✓	-
	Attempt to Access IP-Restricted APIs via Proxy Headers	✓	-
	Attempt to Access Restricted Operations through Allow-Listed Operation Names	✓	-
	Test for GraphQL Authorization Traversals	✓	-
	Test Signature Validation in JWTs	✓	-
	Check for Sensitive Data in JWTs	✓	-
	Test for Brute Force Mutations or Queries that Accept Secrets alias-based query batching	✓	✓
	GraphQL: Injection		
	Identify All Fuzzable Inputs	✓	-
	Test for injection in Query Arguments	✓	✓
	Test for injection in Field Arguments	✓	✓
	Test for injection in Query Directive Arguments	✓	✓
	Test for injection in Operation Names	✓	✓
	Test for SQL Injection	✓	✓
	Test for OS Command Injection	✓	✓
	Test for XSS Injection for HTLM Rendering Calls	✓	✓
	Test for Injection via File Uploads	✓	✓
	GraphQL: Forging Requests		
	Test for existence of anti-CSRF tokens in HTTP Headers or Bodies	✓	-
	Test for validity of anti-CSRF tokens	✓	-
	Test for reuse of anti-CSRF tokens	✓	-
	Test for availability of GET-based queries	✓	-
	Test for support for GET-based mutations	✓	-
	Perform state-changing mutations over GET	✓	-
	Perform state-changing mutations over POST	✓	-
	Test for Semi-Blind SSRF Vulnerabilities	✓	✓
	Test for Blind SSRF Vulnerabilities	✓	✓
	GraphQL: Hijacking Requests		
	Identify whether the GraphQL server support subscriptions	✓	-
	Validates the Origin header during a WebSocket handshake	✓	-
	Test for Cross-Site WebSocket Hijacking	✓	-
	Test for HTTP Cookie Attributes	✓	✓
Exploitation	Test for bypass attacks	✓	✓

Phase	Tasks Completed	Manual	Automated
	Test for injection attacks	✓	✓
	Test for session attacks	✓	✓
	Attempt to escalate privileges and/or gain unauthorized access	✓	✓
	Attempt to pivot from compromised systems to other internal systems.	✓	✓
<b>Reporting</b>	A draft detailed report outlining findings coupled with control recommendations including an executive summary outlining the overall state of the application.		
	Document steps to reproduce findings to ensure application developers can validate remediation efforts prior to retesting.		
	Conduct root cause analysis of findings outlining common themes observed with recommendations to improve security within the environment.		

## 6.2 External Infrastructure Penetration Testing

Our security testing methodology is derived from the SANS Pentest Methodology, the MITRE ATT&CK framework for enterprises, and NIST SP800-115 to ensure compliance with most regulatory requirements. Our methodology is comprehensive and has been broken up based on which areas can be tested with automation and those which require extensive manual testing.

Phase	Tasks Completed	Manual	Automated
<b>Discovery</b>	Comprehensive port scanning, fingerprinting and mapping of services and applications	✓	✓
	Identify password length and lockout policy	✓	✓
	Identify centralized management authentication servers	✓	✓
	Conduct passive network traffic analysis	✓	✓
	Conduct DNS based enumeration	✓	✓
	Identify host-based and network-based defense technologies	✓	✓
	Enumerate infrastructure and admin interfaces	✓	✓
<b>Vulnerability Assessment</b>	Configuration and Deploy Management Testing		
	Utilization of automated scanning tools & technologies to identify publicly known operating system, application and service vulnerabilities	✓	✓
	Manual validation of findings, removal of false-positives, and low-confidence findings where applicable	✓	-
	Test for unencrypted management interfaces and services	✓	✓
	Test for SSL/TLS configuration and certificates	✓	✓
	Test for egress filtering controls	✓	-
	Test for server and endpoint hardening	✓	-
	Test for stale network address configurations	✓	-



Phase	Tasks Completed	Manual	Automated
	Test for anonymous access	✓	✓
	Test for insecure configuration of various TCP and UDP services	✓	✓
	Test for servers and endpoints without endpoint protection	✓	-
	Patch Level Testing		
	Identify false positives through internal and commercial tooling	✓	✓
	Identify and prioritize vulnerable targets for initial access	✓	✓
	Test for missing security patches and EOL software	✓	✓
	Test for common and known vulnerabilities in weak applications	✓	✓
<b>Exploitation</b>	Vulnerabilities and Misconfigurations		
	Exploitation of identified vulnerabilities and misconfigurations using public exploit code, or custom exploits as applicable	✓	✓
	Demonstrate impacts of identified vulnerabilities on the environment, data, services or applications	✓	✓
	Credential Access		
	Test for default credentials on operating systems, services and applications	✓	✓
	Test for insecure credential storage (browser storage, plain-text documents, email, file-share, etc.)	✓	✓
<b>Reporting</b>	A draft detailed report outlining findings coupled with control recommendations including an executive summary outlining the overall state of the application.		
	Document steps to reproduce findings to ensure application developers can validate remediation efforts prior to retesting.		
	Conduct root cause analysis of findings outlining common themes observed with recommendations to improve security within the environment.		

# Ready to strengthen your security posture?

**There's simply no room  
for compromise.**

Get in touch to share your  
cybersecurity needs with our  
team and get a free quote.

☎ 647 797 9230   @ info@packetlabs.net

🌐 packetlabs.net

📍 606-6733 Mississauga Road, Mississauga, ON, L5N 6J5

🐦 @pktlabs   in /packetlabs-ltd-

f @packetlabs

